Hormone Therapy DSS for Breast Cancer

Benjamin Pepper

June 16, 2020

Hormone Therapy DSS for Breast Cancer

The purpose of this project is to aid physicians in deciding whether or not to recommend hormone therapy to breast cancer patients by providing a decision support system. This DSS allows physicians to input 16 clinical values and a genomic score and outputs the probability that a patient received hormone therapy treatment based upon an integrated multi-stage model trained and tested on the BRCA Metabric genomic data set from cBioPortal for Cancer Genomics.

Get Genomic & Clinical Datasets:

The BRCA Metabric genomic data set from cBioPortal for Cancer Genomics was used to obtain genomic and clinical data of breast cancer patients from "data_expression_median.txt", "data_clinical_patient.txt", and "data_clinical_sample.txt" files (https://www.cbioportal.org/study/summary?id=brca_metabric).

Direct Download Link of Data: http://download.cbioportal.org/brca_metabric.tar.gz (Recommend 7-zip for extraction)

The functions called below serve to clean the data for this project and their definitions can be found in "ht_functions.R" along with those of other functions written for this project.The fread() function was used in reading in the data for its superior performance when working with Big Data sets. This is owed to its lazy execution which avoids storing the entire Big Data set in memory, only the parts necessary for the requested operations.

```r
express = get_ht_express_dat(root)
express$y = as.factor(express$'HORMONE_THERAPY')
express = express[,!colnames(express) %in% 'HORMONE_THERAPY']

clin = get_ht_clin_dat(root)
```

Clinical Data Reduction & Dummification:

Utilizing domain knowledge, the variables ['ER_STATUS', 'ER_IHC', 'PR_STATUS', 'HER2_STATUS', 'INTCLUST', 'HORMONE_THERAPY', 'VITAL_STATUS', 'HISTOLOGICAL_SUBTYPE', 'OS_STATUS', 'OS_MONTHS', 'SAMPLE_ID', 'THREEGENE', 'CANCER_TYPE', 'CANCER_TYPE_DETAILED', 'SAMPLE_TYPE'] were removed from the set of clinical predictors because they were determined to either be information collected only after a patient has begun receiving a treatment or are otherwise collinear with the response variable 'HORMONE_THERAPY'

```r
var_remove = c('ER_STATUS', 'ER_IHC', 'PR_STATUS', 'HER2_STATUS', 'HORMONE_THERAPY',
               'INTCLUST', 'VITAL_STATUS', 'HISTOLOGICAL_SUBTYPE', 'OS_STATUS',
               'OS_MONTHS', 'SAMPLE_ID', 'THREEGENE', 'CANCER_TYPE',
               'CANCER_TYPE_DETAILED', 'SAMPLE_TYPE')
clin_reduced = na.omit(clin[,!colnames(clin) %in% var_remove])
clin_rownames = rownames(clin_reduced)
clin_dummies = dummy_cols(clin_reduced, remove_selected_columns = T, remove_first_dummy=T)
rownames(clin_dummies) = clin_rownames

common = intersect(rownames(clin_dummies), rownames(express))
```

```
clin = clin_dummies[common,]
express = express[common,]
clin$y = express$y

dat = list(express, clin)
```

Data Integration:

Multi-stage analysis data integration was performed by training a model on the genomic data set, integrating the results with the clinical daset, and then training a model on the integrated dataset. For the first stage, a lasso penalized regression model was trained on the genomic expression levels of 24,368 genes/features from the Breast Cancer BRCA Metabric genomic data set. The prediction scores were then taken for each observation as a new variable "GENOMIC_SCORE" that was integrated into the dummified clinical data set. For the second stage, a logistic regression was performed on the integrated data set and the prediction scores were interpreted as the probability of a patient receiving hormone therapy. 1900 multi-stage models were trained and tested on subsets of the data after accounting for the 10 nested folds within each of the 10 outer cross-validation folds and 19 different values for lambda. The final model was constructed from the whole data set with the lambda hyperparameter value yielding the highest performance. With 17 features for physicians to input, the model is suitable for use in decision support systems for determining whether or not a breast cancer patient would be a good candidate for hormone therapy.

```
integrated_grid = data.frame(lambda = seq(0.000001, 1, by = 0.05))
res_integrated = nested_cv(cv_k1 = 10, cv_k2 = 10, seed = 1, model = integrated_mod,
                  inner_perf_f = integrated_inner_perf,
                  outer_perf_f = integrated_outer_perf,
                  score_f = integrated_score, perf_type = 'high',
                  grid = integrated_grid, dat = dat, response = 'y')
res_integrated$Performance
mean(res_integrated$Performance$Perf)

best_integrated_params = mean(integrated_grid[res_integrated$Performance$BestParams,])
mod_integrated = integrated_mod(dat, response = 'y', best_integrated_params)
clin_reduced$GENOMIC_SCORE = lasso_score(mod_integrated$mod1, express, response = 'y')[,1]
clin_reduced <- clin_reduced %>% select(GENOMIC_SCORE, everything())
```

Running the DSS Shiny App:

To run the DSS, open "ht_dss.R" in RStudio and click the green arrow and text "Run App" in the top right corner. Be sure to have the .RDS files in the same folder that were saved with the below code:

```
saveRDS(mod_integrated, 'integrated_mod.RDS')
saveRDS(res_integrated$Scores, 'integrated_scores.RDS')
saveRDS(res_integrated$Y, 'integrated_labels.RDS')
saveRDS(clin_reduced, 'integrated_dat.RDS')
```

Performance:

```
mod = readRDS('integrated_mod.RDS')
integrated_results = read.csv("integrated_results.csv")
```

The resulting multi-stage model yields an average accuracy of 77.8% across the 10-folds and was trained with a lambda value of 0.130001. The accuracy for each of the individual 10-folds can be seen below:

```
integrated_results
```

```
##         Perf
## 1  0.7835821
## 2  0.7925926
## 3  0.8148148
## 4  0.7481481
## 5  0.7259259
## 6  0.7761194
## 7  0.8000000
## 8  0.8000000
## 9  0.7313433
## 10 0.8059701
```